California State University Stanislaus
Department of Computer Science
Syllabus

Instructor Dr. Xuejun Liang
My Office: DBH 282
Office Hours: MW 1:00 p.m.-2:00 p.m. & Th 11:00 a.m.-12:00 a.m.
ZOOM Meeting ID: 4438930033, Phone: (209) 667-3169, Email: xliang@csustan.edu

Class Information

Classroom: DBH 103
Class Date & Time: TTh 9:30 a.m. – 10:45 a.m.
Class Website: https://www.cs.csustan.edu/~xliang/Courses2/CS4300-23F
Class Server: wozniak.csustan.edu (You can use PuTTY to remote login.)

Catalog Description

CS4300 Compiler Theory. (3 Units) Pre-requisites: CS 4100. Lexical, syntactic, and semantic analyses and syntax directed translation of programming languages. Includes symbol table construction, error diagnostics, and code generation.

Required Textbook

Compilers: Principles, Techniques, & Tools , 2/E, by Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Addison Wesley, 2007. ISBN: 9-780-321-486-813

Reference Books

1. Compiler Construction: Principles and Practice, by Kenneth C. Louden, PWS Publishing Company, 1997
2. flex & bison: Text Processing Tools, by John Levine, O'Reilly Media, Inc., 2009
3. Modern Compiler Implementation in Java, 2/E, by Andrew W. Appel and Jens Palsberg, Cambridge University Press, 2002
4. Modern Compiler Implementation in C, by Andrew W. Appel and Maia Ginsburg, Cambridge University Press, 2004
5. Advanced Compiler Design and Implementation, by Muchnick Steven, Morgan Kaufmann, 2008
6. Engineering a Compiler, 2nd Edition, by Keith Cooper and Linda Torczon, Addison Wesley, 2011

Handouts: (Available on Class Website)

1. Cool Reference Manual
2. Tour of Cool Support Code
3. Cool Runtime System
4. Cool Example Programs

Reference Materials: (Available on Class Website)

5. Bison manual
6. Java CUP manual

Course Outcomes

Students who successfully complete the course must be able to

1. Build a compiler for a (simplified) programming language.
2. Utilize compiler construction tools, such as generators of scanners and parsers.
3. Apply algorithms for lexical analysis.
4. Apply algorithms for LL(1), LR(1), and LALR(1) parsing.
5. Select an intermediate representation, translate expressions, and check types.
6. Describe how the compiler creates and manages a run-time environment to support execution of its target programs.
7. Describe how the code generator performs instruction selection, register allocation and assignment, and instruction ordering.

Course Outline* (Major Topics and Weekly Schedule)

Date	Topics Covered	
Week 1: 08/22, 08/24	Syllabus and Introduction of the class, The Structure of a Compiler, The Science of Building a Compiler, Compiler Technology, and Programming Language Basics.	Ch1
Week 2: 08/29, 08/31	Introduction to Compiler Frontend: Syntax Definition, Syntax-Directed Translation, and Parsing.	Ch2
Week 3: 09/05, 09/07	Introduction to Compiler Frontend: A Translator for Simple Expressions, Lexical Analysis, Symbol Tables, and Intermediate Code Generation COOL to prepare for PA1	Ch2
Week 4: 09/12, 09/14	Lexical Analysis: Specification of Tokens, Recognition of Tokens, and Lexical-Analyzer Generator: Lex and Flex.	Ch3
Week 5: 09/19, 09/21	Lexical Analysis: Finite Automata, From Regular Expressions to Automata, Construct an NFA from a Lex Program, and From RE to DFA Directly Using lex/flex to prepare for PA2	Ch3
Week 6: 09/26, 09/28	Syntax Analysis: Context-Free Grammars, Writing a Grammar, Top-Down Parsing, and Bottom-Up Parsing	Ch 4
Week 7: 10/03, 10/05	Syntax Analysis: Simple LR and More Powerful LR Parsers. Midterm Exam	Ch4
Week 8: 10/10, 10/12	Syntax Analysis: Using Ambiguous Grammars, Parser Generators Using yacc/bision to prepare for PA3	Ch4
Week 9: 10/17, 10/19	Syntax-Directed Translation: Syntax-Directed Definitions and Evaluation Orders for SDD's	Ch5
Week 10: 10/24,	Syntax-Directed Translation: Applications of Syntax-Directed Translation and Syntax-Directed Translation Schemes.	Ch5

10/26		
Week 11: 10/31, 11/02	Semantic Analysis: Types and Declarations and Type Checking Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code	Ch6
Week 12: 11/07, 11/09	Intermediate-Code Generation: Translation of Expressions, Control Flow, and Backpatching	Ch6
Week 13: 11/14, 11/16	Run-Time Environments: Storage Organization, Stack Allocation of Space, and Access to Nonlocal Data on the Stack	Ch7
	Thanksgiving	
Week 14: 11/28, 11/30	Code Generation: Issues in the Design of a Code Generator, The Target Language, and Addresses in the Target Code	Ch8
Week 15: 12/05, 12/07	Code Generation: Basic Blocks and Flow Graphs Review for the Final Exam	Ch8
Week 16: 12/12, 12/14	Final Examination Schedule https://www.csustan.edu/class-schedule/finals-schedule	

*It is subject to change.

Grading Scale will be assigned on a standard scale as below:

A	B	C	D	F
90-100	75-89	60-74	45-59	<45

Clustering of grades may cause the grading scale to be lowered (to your benefit), but it will not be raised.

Evaluation

The overall course grade will be the weighted sum of the points earned in the following categories:

Participation	Homework	Projects	Midterm Exam	Final Exam
10%	20%	15%	25%	30%

Other Polices

1. I will accept late assignments for a maximum of three days (including holidays) with a point deduction of 20% per day.
2. There will be no makeup exams except in a verified emergency with immediate notification.

Academic Honesty

The work you do for this course will be your own, unless otherwise specified. You are not to submit other people's work and represent it as your own. I consider academic honesty to be at the core of the University's activities in education and research. Academic honesty is always expected in this course.

Accommodations for Students with Disabilities

Students with disabilities seeking academic accommodations must first register with the Disability Resource Services (DRS) program, located in MSR 210, ph. (209) 667-3159. Students are encouraged to talk with the instructor regarding their accommodation needs after registering with DRS.